Empowerment-Driven Single Agent Exploration For Locating Multiple Wireless Transmitters

Daniel Barry, Andreas Willig, and Graeme Woodward

dan.barry@pg.canterbury.ac.nz, andreas.willig@canterbury.ac.nz,
graeme.woodward@canterbury.ac.nz
University of Canterbury, Christchurch 8140, New Zealand

Abstract. Unmanned Aerial Vehicles (UAVs) have attracted significant interest in recent years, as they have shown to be effective in supporting a wide range of applications in many different areas, including logistics, search and rescue (SAR) [3], public safety communications, [8], infrastructure monitoring [9], precision agriculture [4], forestry [5], and telecommunications [2]. Specifically we focus on those of search and exploration in the context of search and rescue. In our presented work, success is measured in an agents ability to find all transmitters in as small a time as possible. Through the use of a challenging discretized simulation environment, we investigate the practicality of an empowerment-driven exploration behaviour (EEB) in order to locate an unknown number of wireless transmitters with minimal prior knowledge about the locations of obstacles, transmitters and their properties. With problem specific adaptations to the algorithm, including the ability to detect non-identifying signals from transmitters, when compared with a random walk agent and an idealistic Bayesian agent, the empowerment algorithm performs near to that of the Bayesian agent with unrealistic information about the environment. We show that our empowerment-driven algorithm has practical potential and lays a foundation for future work in this area.

Keywords: Empowerment · Search and Rescue · Wireless Transmitters

1 Introduction

We are interested in SAR operations and in particular on a scenario where several people carrying some kind of wireless transmitter (e.g. their cellphone, laptop, smart watch) are distributed over a fixed area and need to be located so that they can receive rescue assistance. We assume that their wireless transmitters frequently send out some signal, although the period is unknown. The overall aim is to minimize the (average) time required to detect and localize all wireless transmitters, assuming that an increased time (cost) taken to find these targets has negative consequences [1].

We have chosen to investigate the viability of empowerment [7] to drive the behaviour of UAV agents for SAR. Bayesian search models have been proven effective in time-critical SAR operations, but there are still open questions about path planning [10]. Empowerment offers an intrinsic motivation for agents to search an environment, offering the ability to negotiate immediate loss of "reward" in favour of

long-term opportunity to discover a transmitter. The empowerment-based algorithm developed in this paper comes with $O(|A|^N)$ time complexity, where $|A|$ is the number of possible actions and N is the look-ahead variable. We compare the performance of this algorithm against two baseline schemes and find that it offers detection times much shorter than a random search and competitive with the times achievable with an idealized Bayesian agent already knowing the environment.

2 Background

A	Set of possible action states	L	Length of the environment in patches
B	Set of obstacles within the environment	n	Number of empowerment steps
C	Channel capacity	S	Set of possible sense states
ζ	Channel capacity with prediction decay	t	Discrete time step for the environment
$\mathfrak{E}$	Empowerment value	T	Set of transmitters
$f(\cdot)$	Function for calculating agent action	W	Set of possible world states
$g(\cdot)$	Function for internal agent update	λ	Information decay value

Table 1: Nomenclature for equations in this paper.

2.1 Perception Action Loop

We model our agent in a discrete-time perception action loop as shown in Figure 1. At each time step (or tick), t, the real world is in state W_t. An agent (i.e. a UAV) is given sensor input S_t, updates its own internal world model to become the new internal model M_t, and then calculates an action to be carried out, A_t, which is taken from a finite set of actions available in the current internal state M_t. The action taken in turn has an impact on the real world state, which changes to become W_{t+1} at the start of the next round. An agent essentially wants to choose its action A_t so that it maximizes its chances of detecting or even localizing a transmitter. In picking its action A_t the agent can choose to consider the consequences of the actions into the future, for example over a time horizon of the next n steps (lookahead). The agent first updates its internal model using the behaviour g, i.e. $M_t = g(M_{t-1}, S_t)$, and then calculates its best action using behaviour f, i.e. $A_t = f(M_t, n)$. In this paper we look to define suitable representations for the sensor data S_t, the internal model M_t and the two behaviours $g(\cdot)$ and $f(\cdot)$.

2.2 Empowerment

Empowerment is an information-theoretic algorithm that describes the control an agent has over its environment (whether this is actual control or perceived control from an internal model) in the perception-action loop (see Section 2.1). Empowerment can also be interpreted as allowing an agent to estimate how much control it has and

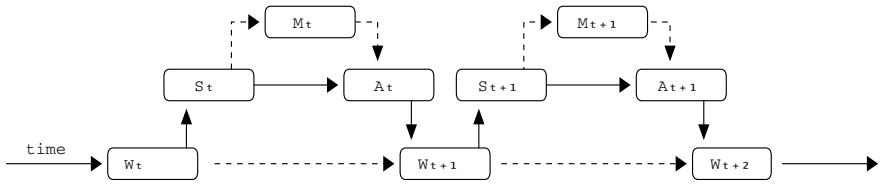


Fig. 1: Agent's perception action loop with memory.

to choose its actions so as to maximize it's capability of maintaining many control options in the future [6]. In other words, an agent driven by empowerment aims to "keep its options" as open as possible. When applied to the SAR problem, we interpret the notion of "option" or "control" here by the opportunities to discover a transmitter.

With one-step empowerment we aim to choose our action A_t to maximize our information about the location of transmitters in the next step, i.e. to maximize our chances of getting the desired sensor inputs S_{t+1}:

$$\mathfrak{E}_1 := C(A_t \rightarrow S_{t+1}) \equiv \max_{p(a_t)} I(S_{t+1}; A_t) \tag{1}$$

With n-step empowerment we aim to choose A_t to maximize our information about the location of transmitters within the next n steps:

$$\mathfrak{E}_n := C(A_t \rightarrow S_{t+n}) \tag{2}$$

3 System Model and Evaluation Method

3.1 Environment

We assume that the search environment W (i.e. the pre-defined area within which to search for transmitters) is two-dimensional and has the shape of a square, with sides of length L. The obstacles are placed randomly, with a given probability $p(B)$ of finding an obstacle within a patch.

We have used an algorithm from maze design, particularly we are using a depth-first search (recursive-stack) backtracker to place obstacles. Obstacles are then randomly removed until the desired ratio of obstacles in the environment is obtained - whilst maintaining a fully explorable environment. Varying the number of obstacles in the environment changes the scenario difficulty.

3.2 Wireless Transmitters

The transmitters are randomly placed into the environment, particularly into patches without obstacles, such that no two transmitters are in the same patch. We choose a uniform distribution for placing transmitters into patches.

In this paper we use the simple *unit disc model* for transmitter detection. In this model there is given a radius around the transmitter. If the receiver is within this radius, a signal is received with 100% probability and if outside this radius, signal is 0% probability. A transmitter transmits signals periodically from a uniform distribution between 4 ticks and 10 ticks and do not contain any information allowing the UAV to uniquely identify the transmitter, the UAV can only tell whether a signal is detected or not. In our model signals do not overlap or interfere with one another.

3.3 State, Sensing and Action Spaces

The world state is given by a vector $(W_{x,y}: x,y \in \{1,...,L\})$ with one state value $W_{x,y}$ for each patch (x,y). The patch occupancy is given by $W_{x,y} \in \{EMPTY, OBSTACLE, TRANSMITTER\}$.

With respect to sensing we make the following assumptions:

- The UAV agent has a GPS facility and can always tell with certainty in which patch (x,y) it currently is. The UAV is restricted to being in patches without obstacles.
- The UAV agent has a downward-facing camera, which allows to determine with certainty whether a transmitter is directly below the UAV agent or not. A transmitter in square (x,y) is detected with the downward camera only when the UAV position is (x,y), too.
- The UAV has further sensors allowing it to determine whether the eight neighboured patches contain obstacles (with obvious adaptations if the UAV is at the boundary of the environment). This is called the Moore neighbourhood.

All these quantities are being made available to the UAV as the sensing input S_t at the start of a tick. In addition there is the input from the radio receiver, which the agent receives while being in the current patch. The action space of the UAV agent reflects its options for movement, more precisely, when the agent is in patch (x,y) it gives the possible movements into any neighboured patch for the next tick, taken from the set $\mathcal{A} = \{\text{NORTH, EAST, SOUTH, WEST}\}$.

3.4 Performance Measure

We vary both the number of transmitters and the density of obstacles independently and record the average time to accurately detect all transmitters. The simulation keeps track of transmitters the agent has accurately located by visiting them (i.e. the agent being in the same patch and detecting the transmitter with the downward camera). The average is taken over a number of realizations of the maze.

4 Comparison Algorithms

The **random walk** algorithm is very simple and can be considered a lower bound. It does not keep any internal state (i.e. the state update function $g(\cdot)$ is empty) and it selects the next patch randomly with uniform distribution (it can tell which of the neighbouring patches is admissible based on sensor inputs, it does not need to keep track of the environment).

We also use a **Bayesian search** algorithm which we expect to perform quite well, by virtue of already having a-priori information about the environment, which EEB does not have. Particularly, the Bayesian search algorithm knows a-priori which patches contain obstacles and which ones don't. The location of transmitters is not known to the Bayesian search agent. Intuitively, the behaviour of the Bayesian search agent is always to go next to the nearest patch which it has not yet visited, this way exhausting all non-obstacled patches in a greedy fashion.

5 Empowered Exploration Behaviour (EEB)

5.1 Algorithm Overview

Building on empowerment, our agent employs a few key differences:

- A preference for information in the near future: suppose the agent considers two alternative paths of n steps each, and both with the same number of yet-unexplored patches, i.e. both allowing for the same information gain. According to the definition of empowerment both possible paths would be of the same value, but in our algorithm we give preference to the path which leads more quickly to expected information gain.
- The use of transmitter signals to prioritize search, i.e. when the agent receives transmitter signals in its current patch, it gives preference to close-by patches in order to quickly locate the transmitter(s) currently close to it. With this, the agent spends more time searching a given area with the expectation the signal may reveal a new transmitter.
- When there exists two or more actions of the same maximal empowerment value, we use a further heuristic to break the ties, where preference is given to the option that leads to a newly discovered transmitter with a higher probability. This is explained in more detail below.

5.2 Internal Memory

The EEB agent maintains an internal state M_t which is updated from the sensed information about the environment. More precisely, to each patch (x,y) the agent associates the following information:

- A belief value $W_{b(x,y)}$ which encodes the current knowledge of the agent about this patch, giving the probability of an undiscovered transmitter existing. As $W_{b(x,y)} \to 0$, the probability of discovering a new transmitter is low and at 0 the agent has directly observed the patch and confirmed there is no transmitter. As $W_{b(x,y)} \to 1$, the probability of discovering a new transmitter is high. The agent operates under the assumption that there is always a new transmitter to be found, which cannot be confidently proved true or false until the entire environment is searched.
- Whether a patch can be explored (because of an obstacle) is stored in $W_{e(x,y)}$, where by default 1 indicates the state is explorable until an observation suggests otherwise, in which case $W_{e(x,y)} = 1$.
- The number $W_{s(x,y)}$ of radio signals heard while being on this patch: this is a counter incremented each time the agent is in this patch and hears a wireless signal.

– The location of transmitters found $W_{t(x,y)}$, 0 by default indicates no transmitter located, whereas 1 indicates a transmitter found.

Besides this information the agent knows its own position at any time, represented as the patch (x,y) it is currently in.

5.3 Update Function $g(\cdot)$

In each tick, we update our internal model M depending on the sensor input S as follows, assuming the agent is currently in patch (x,y):

1. *Increment signal reception counters*: when the agent has heard a signal while being in patch (x,y) the counter $W_{s(x,y)}$ is incremented according to the number of signals overheard.
2. *Record when no transmitter found*: When the downward sensor in the current location (x,y) indicates the absence of a transmitter we assign the belief value $W_{b(x,y)}=0$.
3. *Record when transmitter found*: When the downward sensor in the current location (x,y) indicates the presence of a transmitter we assign the belief value $W_{t(x,y)}=1$.
4. *Updating belief about neighboured patches*: the agent uses its further sensors to check neighbouring patches for the presence of obstacles. If, while the agent is in patch (x,y) these sensors indicate an obstacle in a neighboured patch (u,v), then we update the belief value $W_{e(u,v)}=0$ and by extension, a transmitter may not exist and $W_{b(u,v)}=0$.
5. *Keep track of update rates*: whenever we update any part of our internal model during a tick, we increment the counter c by one. When calculating $1-(c/ticks)$, we can calculate the average probability ρ) under the assumption this invalidates our previous empowerment calculations of the world. As time passes, ρ will converge to 0, and we use ρ as a discount factor when weighing information gain on n steps.
6. *Identify areas with unaccountable signal(s)*: we consider two scenarios: (i) the detected signal at $W_{s(u,v)}$ is within radio range of a transmitter $W_{t(i,j)}=1$ and our current model is $W_z=W_b$, (ii) the detected signal cannot be accounted for, in which case $W_{z(i,j)}$ is $W_{b(i,j)}$ times the sum of all local signals $W_{s(u,v)}$ divided by the total number of unexplored patches when the sum is not zero.

W_z is finally normalized. It is recalculated per tick and represents a heuristic, where larger values indicate a transmitter is more likely.

5.4 Action Function $f(\cdot)$

After the update function $g(\cdot)$, we compute an output A by performing a calculation on our internal model M.

As described by the empowerment Equation 1, we probe actions A for our model W_z, and measure the resulting S to calculate the maximum expected information gain. To perform n-step as seen in Equation 2, for each probed W_z, we perform this step again until n steps deep, choosing the action with the greatest expected information gain.

An exception to this process is that when calculating maximum mutual information for channel capacity, C, we decay this value for the current n-step value. The

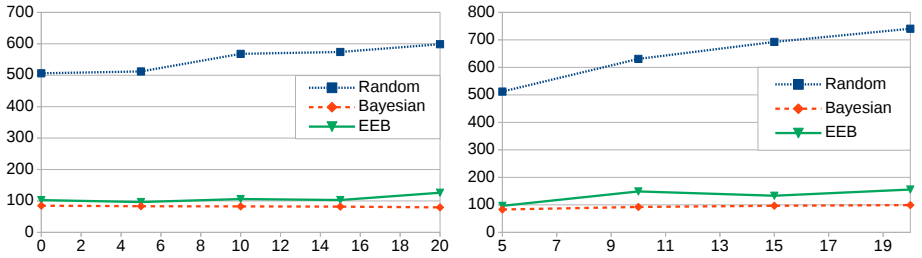
purpose of ζ is to apply a self inflicted cost function to favour near-future expected information gain. We consider the observed model update-rate as a approximation of model accuracy.

$$\zeta = \max_{p(a)} I(S;A).\rho^{n-1} \tag{3}$$

Finally, if our empowerment calculation yields no bias between two or more actions, we sum the probabilities represented by the competing actions and use the largest in order to attempt to split the tie: North: $\sum_{i=0}^{L}\sum_{j=0}^{y+1} W_{z=i,j}$, East: $\sum_{i=x}^{L}\sum_{j=0}^{L} W_{z=i,j}$, South: $\sum_{i=0}^{L}\sum_{j=y}^{L} W_{z=i,j}$, West: $\sum_{i=0}^{x+1}\sum_{j=0}^{L} W_{z=i,j}$. If still no clear action exists, one is randomly selected from the empowerment calculation stage.

6 Results

We have developed a simulator in Java for the purpose of a controlled comparison. For both the random and the Bayesian search algorithm we run 1,000 replications for each considered combination of parameters, where for each replication a new scenario is generated randomly. For EEB we have used >50 averaged replications per parameter combination, due to the computational complexity of this algorithm. The results for the first set of experiments are shown in Figure 2a, and Figure 2b



(a) % of obstacles {0,5,10,15,20,25} vs average ticks (y axis), transmitters = 5.

(b) % of transmitters $\{5, 10, 15, 20, 25\}$ vs average ticks (y axis), obstacles = 5.

Fig. 2: Comparison of algorithms where: (Length) $L=10$, n-step $=12$, transmitter radius of 4 with periods of 4 to 10 ticks.

shows the results for the second set of experiments. We see that the EEB agent was easily able to outperform the random walk agent and generally performs close to the advantaged Bayesian search agent. Interestingly, in the second experiment the gap between EEB and Bayesian search widens somewhat as the number of transmitters is increased. We explain this by our heuristic to not look in the vicinity of already detected transmitters, which can have a tendency to mask further transmitters close to already detected ones. With the exception of the random algorithm, the probability

of obstacles had no measurable effect on the average performance, meaning that the EEB agent was able to successfully navigate around obstacles to find transmitters despite no prior knowledge of where obstacles were placed.

7 Conclusions

The EEB agent appears to be a practical algorithm which can find wireless transmitters efficiently while simultaneously mapping the environment. We see the EEB algorithm as a promising stepping stone towards the development of more refined and more realistic single-agent algorithms, but more importantly we also expect that it can be fruitfully carried over to the case where several agents are used in parallel and are allowed to collaborate with each other, e.g. by sharing belief and counter information. The EEB algorithm is an important step towards information-driven search and exploration agents with an unknown number of objectives. More work is required in order to reduce the computational overhead and allowing for real-time application.

References

1. Adams, A., Schmidt, T., Newgard, C., Federiuk, C., Christie, M., Scorvo, S., DeFreest, M.: Search Is a Time-Critical Event: When Search and Rescue Missions May Become Futile. Wilderness and Environment Medicine **18**(2), 95–101 (2007)
2. der Bergh, B.V., Chiumento, A., Pollin, S.: LTE in the Sky: Trading Off Propagation Benefits with Interference Costs for Aerial Nodes. IEEE Communications Magazine **54**(5), 44 – 50 (May 2016)
3. Erdos, D., Erdos, A., Watkins, S.E.: An Experimental UAV System for Search and Rescue Challenge. IEEE Aerospace and Electronic Systems Magazine **28**(5), 32–37 (May 2013)
4. Gevaert, C.M., Suomalainen, J., Tang, J., Kooistra, L.: Generation of Spectral-Temporal Response Surfaces by Combining Multispectral Satellite and Hyperspectral UAV Imagery for Precision Agriculture Applications. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing **8**(6), 3140 – 3146 (Jun 2015)
5. Ghamry, K.A., Kamel, M.A., Zhang, Y.: Cooperative Forest Monitoring and Fire Detection Using a Team of UAVs–UGVs. In: Proc. 2016 International Conference on Unmanned Aircraft Systems (ICUAS). Arlington, USA (Jun 2016)
6. Klyubin, A., Polani, D., Nehaniv, C.: Empowerment: A Universal Agent-Centric Measure of Control. In: 2005 IEEE Congress on Evolutionary Computation. vol. 1, pp. 128–135. IEEE, Edinburgh Scotland (2005)
7. Klyubin, A., Polani, D., Nehaniv, C.: Keep Your Options Open: An Information-Based Driving Principle for Sensorimotor Systems. In: Sporns, O. (ed.) PLoS ONE. vol. 3. PLoS ONE (2008)
8. Merwaday, A., Guvenc, I.: UAV assisted heterogeneous networks for public safety communications. In: Proc. Wireless Communications and Networking Conference Workshops (WCNCW). Istanbul, Turkey (2015)
9. Sa, I., Hrabar, S., Corke, P.: Inspection of Pole-Like Structures Using a Vision-Controlled VTOL UAV and Shared Autonomy. In: Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014). Chicago, Illinois (Sep 2014)
10. Stone, L., Keller, C., Kratzke, T., Strumpfer, J.: Search for the Wreckage of Air France Flight AF 447. Statistical Science **29**(1), 69–80 (2014)